01-20-00

A

THE ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

DOCKET: ST9-99-033
PATENT

jc584 U.S. PTO
09/487401
01/19/00

jc678 U.S. PTO
01/19/00

Sir:

Transmitted herewith for filing is the Patent Application of

Inventor(s): **John R. Shedden**

For: **ACTIVE LOG READ I/O BALANCING FOR LOG DUPLEXING**

Enclosed with the Patent Application are:

- X **Two** sheet(s) of formal drawing(s)
- X A Declaration and Power of Attorney
- X Information Disclosure Statement (PTO Form 1449)
- ___ A certified copy of a _____ application.
- X An assignment of the invention to: **International Business Machines Corporation** including $40.00 recordation fee

The filing fee has been calculated as shown below:

	(Col. 1)	(Col. 2)	OTHER THAN A SMALL ENTITY	
FOR:	NO. FILED	NO. EXTRA	RATE	FEE
BASIC FEE				$690.00
TOTAL CLAIMS	18 - 20	=	total extra x 18 =	$
INDEP. CLAIMS	3 - 3	=	indep extra x 78 =	$
___ MULTIPLE DEPENDENT CLAIM PRESENTED			+260 =	$
*If the difference in Col. 1 is less than "0", enter "0" in Col. 2			TOTAL	**$690.00**

X A check in the amount of **$730.00** to cover the filing fee and assignment recordal is enclosed.

X The Commissioner is hereby authorized to charge payment of the following fees associated with this communication, or credit any overpayment, to Deposit Account 01-0467. A duplicate copy of this sheet is attached.
- X Any additional filing fees required under 37 CFR 1.16.
- X Any patent application processing fees under 37 CFR 1.17.

Respectfully submitted,

David N. Koffsky, Esq. (19,905)
Attorney for Applicant(s)
Ohlandt, Greeley, Ruggiero & Perle
One Landmark Square - Suite 903
Stamford, CT 06901
(203) 327-4500

EXPRESS MAIL CERTIFICATE

I hereby certify that the above paper/fee is being deposited with the United States Postal Service, "Express Mail Post Office to Addressee" service under 37 CFR 1.10, on the date indicated below and is addressed to the Asst. Commissioner for Patents, Washington, D.C. 20231.
"Express Mail" No.: **EL 4 7 0 2 3 2 4 3 5 US**
Date of Deposit: **JANUARY 19, 2000**
Person mailing paper/fee: Lorraine Ronnlund
Signature: Lorraine Ronnlund

ST9-99-033

ACTIVE LOG READ I/O BALANCING FOR LOG DUPLEXING

FIELD OF THE INVENTION

This invention relates in general to computer-implemented database systems, and, in particular, to use of dual copies of logs.

BACKGROUND OF THE INVENTION

Databases are computerized information storage and retrieval systems. A Relational Database Management System (RDBMS) is a database management system (DBMS) which uses relational techniques for storing and retrieving data. Relational databases are organized into tables which consist of rows and columns of data. The rows are formally called tuples. A database will typically have many tables and each table will typically have multiple tuples and multiple columns. The tables are typically stored on direct access storage devices (DASD), such as magnetic or optical disk drives for semi-permanent storage.

A table is assigned to a tablespace. The tablespace contains one or more datasets. In this way, the data from a table is assigned to physical storage on DASD. Each tablespace is physically divided into equal units called pages. The size of the tablespace's pages is based on the page size of the buffer pool specified in the tablespace's creation statement. The

bufferpool is an area of virtual storage that is used to store data temporarily. A tablespace can be partitioned, in which case a table may be divided among the tablespace's partitions, with each partition stored as a separate dataset. Partitions are typically used for very large tables.

RDBMS software using a Structured Query Language (SQL) interface is well known in the art. The SQL interface has evolved into a standard language for RDBMS software and has been adopted as such by both the American National Standards Institute (ANSI) and the International Standards Organization (ISO). The SQL interface allows users to formulate relational operations on the tables either interactively, in batch files, or embedded in host languages, such as C and COBOL. SQL allows the user to modify the data and as the data is being modified, all operations on the data are logged in a log file.

When a primary copy of a database is corrupted, for example, due to a system or media failure, or when a user wants to remove modifications to the data (i.e., back out the changes), the primary copy of the database is recovered by applying changes found in the log to a secondary copy of the database. That is, through use of the log file, the copied data is modified, based on the operations and data found in the log file.

Typically, for database systems, it is recommended that dual copies (duplex) of the log be maintained to

prevent a single point-of-failure. The recommendation is to maintain the secondary copy of the log on a separate I/O path from the primary copy of the log. In the event of a failure reading the primary copy of the log, the secondary copy of the log is used to satisfy the request.

Normally, all log read requests are directed to the primary copy of the log, reserving the secondary copy of the log for error recovery. However, in the case where a system failure has occurred, but the primary copy of the log remains accessible, a number of recovery tasks often run in parallel. Accordingly, it often occurs that there are a number of substantially concurrent requests to the log. This action results in a queuing of the requests and a delay in the recovery action.

It is therefore an object of this invention to remove a system recovery bottleneck that is presented when too many requests to the log are queued.

It is another object of the invention to provide a method for more efficient use of the secondary copy of the log.

SUMMARY OF THE INVENTION

The invention is embodied in a method for enabling improved access to data stored in a log of a computer memory system, wherein the computer memory system has

multiple copies of said log, i.e., a primary log and a secondary log. Each log stores data transactions with a database system stored on the computer memory system. The method initially responds to a process request for access to a log, by determining a parameter indicative of demand for access to one of the copies of the log. If the parameter has reached a threshold value, accessing processes are thereafter distributed between the primary and secondary copies of the log so as to balance the work dispatched to the respective log copies. The invention is implemented by a computer system and may also be incorporated into a memory media device such as a magnetic disk.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG 1 illustrates an exemplary computer hardware environment that implements the present invention.

Fig. 2 is a high level flow diagram that illustrates the method of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In the following description of the preferred embodiment, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration, a specific embodiment in which the invention may be practiced. It is to be understood that other embodiments may be utilized without departing from the scope of the present invention.

Hardware Environment

FIG. 1 illustrates an exemplary computer hardware environment that implements the present invention. Computer system 102 is comprised of one or more processors connected to one or more data storage devices 104 and 106 that store one or more relational databases, such as a fixed or hard disk drive, a floppy disk drive, a CDROM drive, a tape drive, or other device.

Operators of computer system 102 use a standard operator interface 108, such as IMS/DB/DC®, CICS®, TSO®, OS/390®, ODBC® or other similar interface, to transmit electrical signals to and from computer system 102 that represent commands for performing various search and retrieval functions, termed queries, against the databases. In the present invention, these queries conform to the Structured Query Language (SQL) standard, and invoke functions performed by Relational DataBase Management System (RDBMS) software.

The SQL interface allows users to formulate relational operations on the tables either interactively, in batch files, or embedded in host languages, such as C and COBOL. SQL allows the user to modify the data. In the preferred embodiment of the present invention, the RDBMS software comprises the DB2® product offered by IBM for UDB ("Universal Database") for the OS/390® operating system. Those skilled in the art will recognize, however, that the present invention has application to any RDBMS software, whether or not the RDBMS software uses SQL.

As illustrated in FIG. 1, the DB2® system for UDB for the OS/390® operating system includes three major components: Internal Resource Lock Manager (IRLM) 110, Systems Services module 112 and Database Services module. IRLM 110 handles locking services for the DB2® system, which treats data as a shared resource, thereby allowing any number of users to access the same data simultaneously. Thus concurrency control is required to isolate users and to maintain data integrity.

Systems Services module 112 controls the overall DB2® execution environment, including managing log data sets 106, gathering statistics, handling startup and shutdown, and providing management support. At the center of the DB2® system is Database Services module 118 that contains plural submodules, including Relational Database System (RDS) 116, Data Manager 118, Buffer Manager 120, Logging System 122 and other

components 114 such as an SQL compiler/interpreter. These submodules support the functions of the SQL language, i.e. definition, access control, interpretation, compilation, database retrieval, and update of user and system data.

The present invention is generally implemented using SQL statements executed under the control of Database Services module 118. Database Services module 118 retrieves or receives the SQL statements, wherein the SQL statements are generally stored in a text file on data storage devices 104 and 106 or are interactively entered into computer system 102 by an operator at monitor 124 via operator interface 108. Database Services module 118 then derives or synthesizes instructions from the SQL statements for execution by computer system 102.

Generally, the RDBMS software, the SQL statements, and the instructions derived therefrom, are all tangibly embodied in a computer-readable medium, e.g. one or more of the data storage devices 104 and 106. Moreover, the RDBMS software, the SQL statements, and the instructions derived therefrom, are all comprised of instructions which, when read and executed by computer system 102, cause computer system 102 to perform the steps necessary to implement and/or use the present invention. Under control of an operating system, the RDBMS software, the SQL statements, and the instructions derived therefrom, may be loaded from data

storage devices 104 and 106 into a memory of computer system 102 for use during operations.

Thus, the present invention may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" (or alternatively, "computer program product" or "memory media") as used herein is intended to encompass a computer program accessible from any computer-readable device, carrier, or media. Of course, those skilled in the art will recognize many modifications may be made to this configuration without departing from the scope of the present invention.

Active Log I/O Balancing for Log Duplexing

The present invention provides a logging system 122 for reducing I/O device contention for the primary copy of a log. The invention reduces the time required to recover from media or system failures.

In some recovery scenarios (e.g., recovering from a database media failure), there may be a number of processes concurrently reading from the primary copy of the log. This may generate I/O contention for the device(s) containing the primary copy of the log. As indicated above, this will extend the recovery period.

Logging system 122 relieves some of the device contention by satisfying a portion of the log read requests from the secondary copy of the log in periods of concurrent log read activity. In the embodiment illustrated in the logic flow diagram of Fig. 2, logging system 122 maintains a count of active processes reading the log (step 200). If the count is below a threshold count, logging system 122 assigns a preference for the primary copy of the log to new processes requesting data from the log (step 202).

If the threshold count is reached, logging system 122 distributes preference assignments between the primary copy of the log and the secondary copy of the log, in respect of new processes requesting data from the log (step 204). It is preferred that the preference assignments be alternated as between the log copies so as to balance the work of the respective logs and reduce possible delay times. For a process assigned a preference for the respective copy of the log, log read requests are satisfied from the respective copy of the log (step 206). This action reduces queuing time and quickens system recovery time.

The foregoing description of the preferred embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. For example, instead of keeping count of processes

attempting to access the primary log, a count of requests that have been queued to the primary log may be used. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

Docket Number: ST9-99-033

WHAT IS CLAIMED IS:

1. A method for enabling improved access to data stored in a log of a computer memory system, said computer memory system having multiple copies of said log comprising a primary log and a secondary log, each log storing data transactions with a database system stored on said computer memory system, the method comprising the steps of:

a) responding to a process request for access to a log, by determining a parameter indicative of demand for access to one of said copies of said log; and

b) assigning the process to another of said copies of the log if said parameter has reached a threshold value.

2. The method as recited in claim 1, wherein said one of said copies of the log is the primary log.

3. The method as recited in claim 2, wherein said parameter is a count of the processes assigned to the primary log.

4. The method as recited in claim 3 wherein, when said count of processes assigned to the primary log reaches a predetermined threshold, step b) distributes new process assignments to both the primary log and

secondary log in an attempt to balance work of the respective logs.

5. The method as recited in claim 3 wherein, when said count of processes assigned to the primary log reaches a predetermined threshold, step b) alternates new process assignments to the primary log and the secondary log in an attempt to balance work of the respective logs.

6. The method as recited in claim 2, wherein said parameter is a count of requests that have been queued to the primary log.

7. A memory media including instructions for controlling a computer to enable improved access to data stored in a log of said computer's memory system, said memory system having multiple copies of said log comprising a primary log and a secondary log, each log storing data transactions with a database system stored on said memory system, the memory media comprising:

a) means for controlling said computer to respond to a process request for access to a log, by determining a parameter indicative of demand for access to one of said copies of said log; and

b) means for controlling said computer to assign the process to another of said copies of the log if said parameter has reached a threshold value.

8. The memory media as recited in claim 6, wherein said one of said copies of the log is the primary log.

9. The memory media as recited in claim 7, wherein said parameter is a count of the processes assigned to the primary log.

10. The memory media as recited in claim 8, wherein when said count of processes assigned to the primary log reaches a predetermined threshold, means b) controls said computer to distribute new process assignments to both the primary log and secondary log in an attempt to balance work of the respective logs.

11. The memory media as recited in claim 8, wherein when said count of processes assigned to the primary log reaches a predetermined threshold, means b) controls said computer to alternates new process assignments to the primary log and the secondary log in an attempt to balance work of the respective logs.

12. The memory media as recited in claim 7, wherein said parameter is a count of requests that have been queued to the primary log.

13. A computer system that enables improved access to data stored in a log of said computer's memory system, said memory system having multiple copies of said log comprising a primary log and a secondary log, each log storing data transactions with

a database system stored on said memory system, the computer system further comprising:

a) means for determining a parameter indicative of demand for access to one of said copies of said log; and

a) logging means responsive to a process request for access to a log, by assigning the process to another of said copies of the log if said parameter has reached a threshold value.

14. The computer system as recited in claim 13, wherein said one of said copies of the log is the primary log.

15. The computer system as recited in claim 13, wherein said parameter is a count of the processes assigned to the primary log.

16. The computer system as recited in claim 13, wherein said logging means, when said count of processes assigned to the primary log reaches a predetermined threshold, distributes new process assignments to both the primary log and secondary log in an attempt to balance work of the respective logs.

17. The computer system as recited in claim 13, wherein said logging means, when said count of processes assigned to the primary log reaches a predetermined threshold, alternates new process

assignments to the primary log and the secondary log in an attempt to balance work of the respective logs.

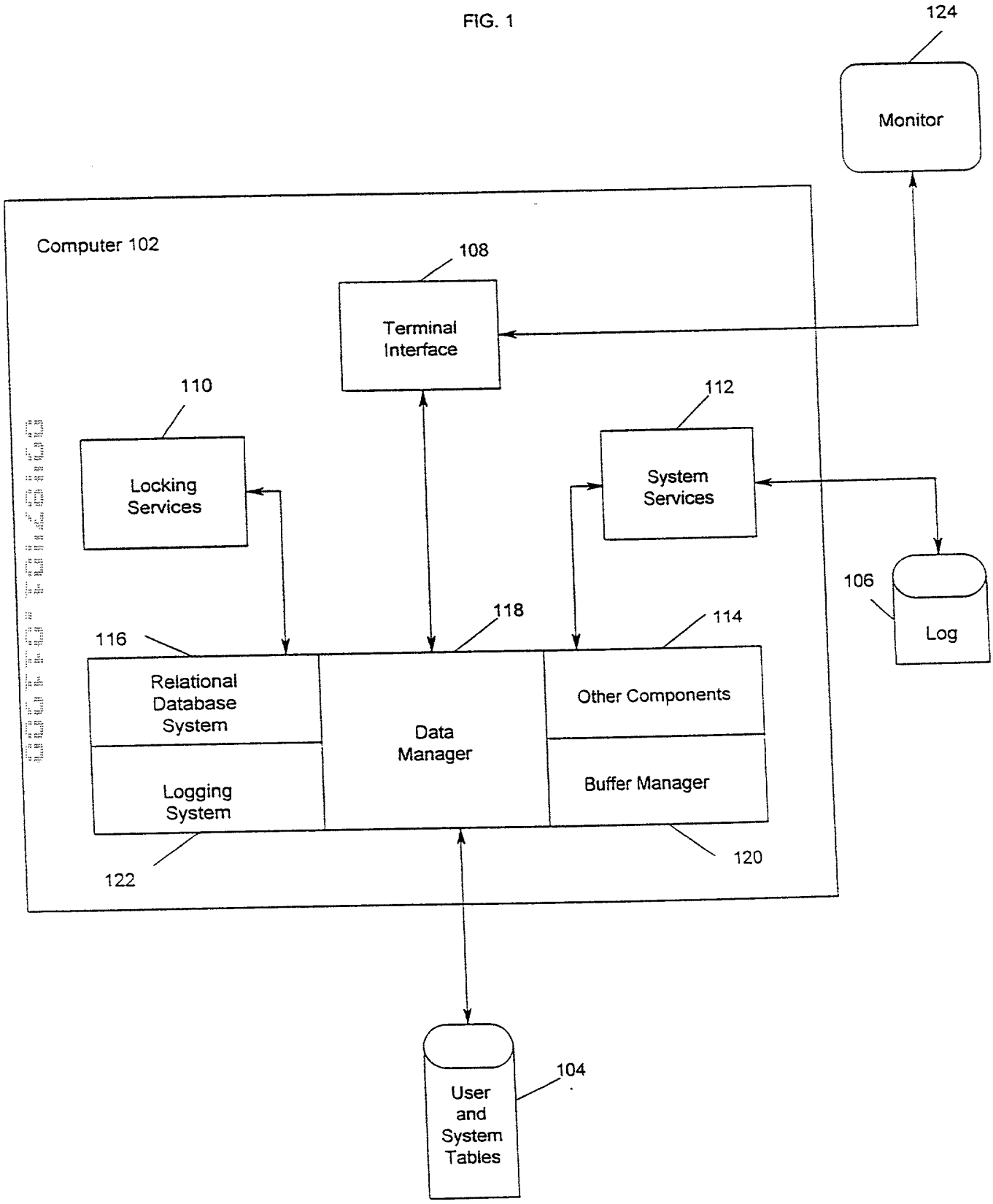
18. The computer system as recited in claim 13, wherein said parameter is a count of requests that have been queued to the primary log.

COPY FURNISHED

ABSTRACT

The invention is embodied in a method for enabling improved access to data stored in a log of a computer memory system, wherein the computer memory system has multiple copies of said log, i.e., a primary log and a secondary log. Each log stores data transactions with a database system stored on the computer memory system. The method initially responds to a process request for access to a log, by determining a parameter indicative of demand for access to one of the copies of the log. If the parameter has reached a threshold value, accessing processes are thereafter distributed between the primary and secondary copies of the log so as to balance the work dispatched to the respective log copies. The invention is implemented by a computer system and may also be incorporated into a memory media device such as a magnetic disk.

FIG. 1


124
Monitor
Computer 102
108
Terminal Interface
110
Locking Services
112
System Services
106
Log
118
114
116
Relational Database System
Data Manager
Other Components
Logging System
Buffer Manager
120
122
104
User and System Tables

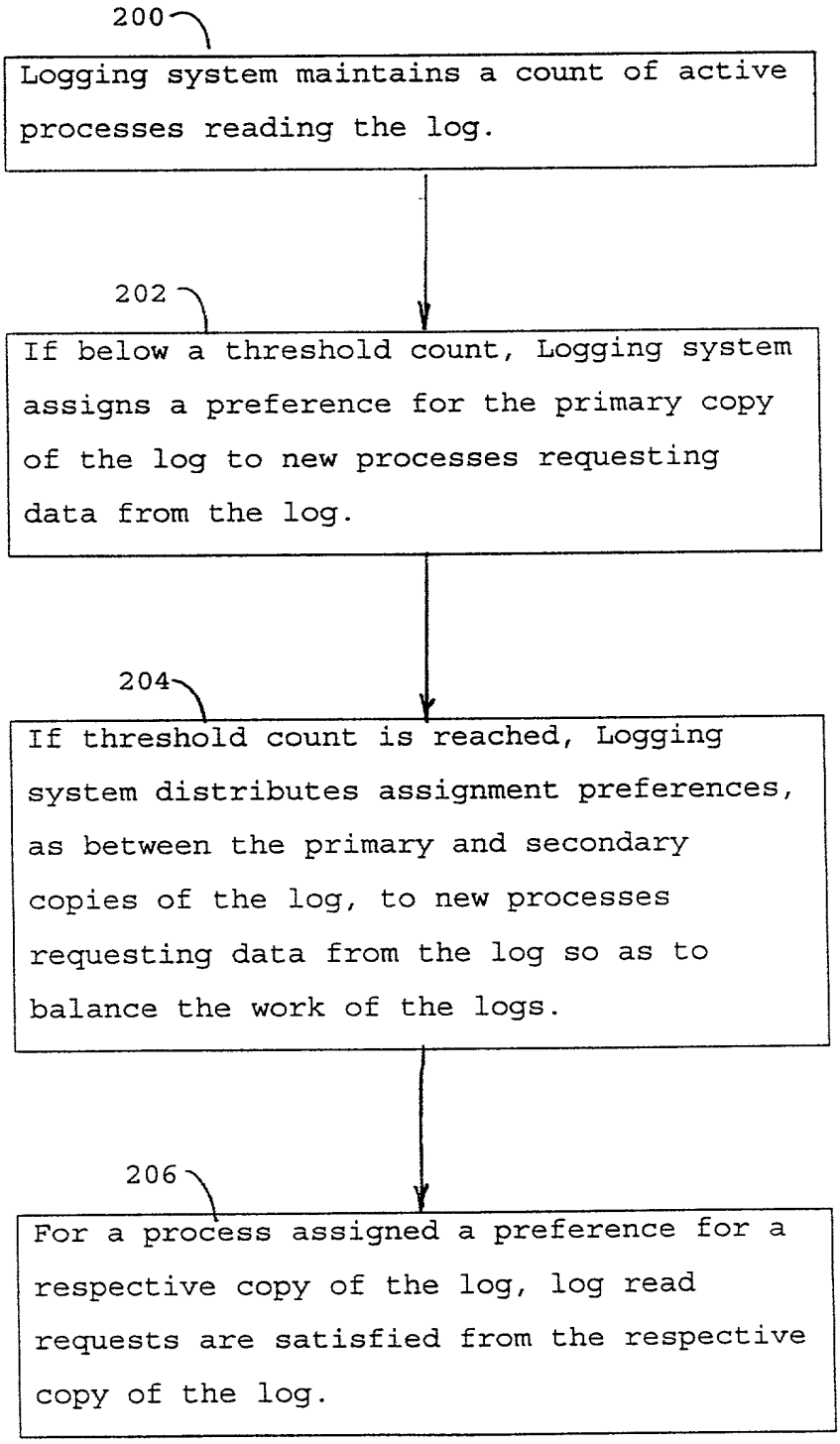

200
Logging system maintains a count of active processes reading the log.
202
If below a threshold count, Logging system assigns a preference for the primary copy of the log to new processes requesting data from the log.
204
If threshold count is reached, Logging system distributes assignment preferences, as between the primary and secondary copies of the log, to new processes requesting data from the log so as to balance the work of the logs.
206
For a process assigned a preference for a respective copy of the log, log read requests are satisfied from the respective copy of the log.

Fig. 2

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
DOCKET: ST9-99-033

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

ACTIVE LOG READ I/O BALANCING FOR LOG DUPLEXING

the specification of which (check one)

__X__ is attached hereto.
_____ was filed on _____
as Application Serial No. _____
and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)
Priority Claimed

None			
___Yes ___No	(Number)	(Country)	(Day/Month/Year Filed)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

None		
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
DOCKET: ST9-99-003

statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Ingrid M. Foerster Reg. No. 36,511	Christopher A. Hughes Reg. No. 26,914	Joseph C. Redmond, Jr. Reg. No. 18,753
Romualdas Strimaitis Reg. No. 35,697	John E. Hoel Reg. No. 26,279	Prentiss W. Johnson Reg. No. 33,123
Timothy M. Farrell Reg. No. 37,321	Edward A. Pennington Reg. No. 32,588	David N. Koffsky Reg. No. 19,905

Send correspondence to:

David N. Koffsky, Esq.
Ohlandt, Greeley, Ruggiero & Perle
One Landmark Square - Suite 903
Stamford, CT 06901

Direct Telephone Calls to:

David N. Koffsky
(203) 327-4500

Full name of inventor: Shedden, John R.

Inventor's signature: John R Shedden Date: 1/5/00

Residence: **16553 Mira Bella Place, Morgan Hill, California 95037**

Citizenship: **U.S.A.**

Post Office Address: Same